

REMARKS

Claims 1, 3-10, 16-26, 28-41, 43-47, 49, 50, and 53 remain pending in the instant application. All pending claims presently stand rejected. Claims 1, 16, 18, 21, 32-41, 45-47, 50, and 53 are amended herein. Entry of this amendment and reconsideration of the pending claims are respectfully requested.

Response to Advisory Action

In the Advisory Action mailed June 11, 2007, the Examiner indicated that a further search is required. Accordingly, a Request for Continued Examination is submitted herewith.

Firstly, in the Advisory Action, the Examiner stated,

Goebel is capable of re-optimizing a binary executable. Re-optimizing compiler process source information by a compiler front-end segment that generates an intermediate representation of the source information.

While Applicants do not agree with this statement, this statement does not address the specific elements of independent claims 1, 18, 32, 41, and 50. In particular, these claims recite that intermediate code is executed based on external execution input. Goebel fails to disclose executing the intermediate representation (IR) code output from binary disassembler 311 or output from compiler front end 303. In contrast, Goebel discloses executing binary executable 313 to generate profile information 317. Executing binary executable 313 does not disclose, teach, or suggest executing the IR.

Secondly, in the Advisory Action, the Examiner stated,

Goebel specifically discloses front end code generator which generates intermediate code based on the source code as cited by the Examiner, see FIG. 3, element 303.

While Applicants agree that Goebel discloses that compiler front end 303 generates intermediate code based on source information 301, this does not address the specific elements of independent claim 35. In particular, claim 35 recites,

the back-end code generator to provide the **modified machine code** to the profiler **for generation of a second data file**;

the **front-end code generator to receive the second data file and produce second intermediate code based directly on the source code and based on the second data file**; and

Goebel discloses that the **first instance** of intermediate code and binary module 309 are generated based on source information 301; however, **subsequent re-optimizations of binary executable 313** do not re-optimize their corresponding intermediate representations based **directly** on source information 301. Rather, FIG. 4 illustrates that the re-optimization of binary executable 313 is based upon the IR output from disassembler 311 and upon the profile information 317 generated by executing binary executable 313—not directly based on source information 301.

The Examiner states

The intermediate representation optimizer segment and the code generator segment also propagate portions of their internally collected symbol and alias information as annotations to the resulting binary module (and corresponding binary executable). This annotation information is used by a binary re-optimization process (subsequently described with respect to FIG. 4) to approximate the compiler's internal state during the compilation of the source code used to create the binary executable (col. 6, lines 22-43).

However, propagating **symbol and alias information as annotations** to be used by the **re-optimization** process does not disclose, teach, or suggest propagating the source code itself. Claim 35 recites, “the front-end code generate to ... produce second intermediate code based **directly on the source code** and based on the second data file. Goebel does not teach or suggest **re-optimizing** based directly on source code; rather, as acknowledged by the Examiner, Goebel discloses **re-optimization** based on annotations including symbol and alias information—not source code.

Claim Rejections – 35 U.S.C. § 101

Claims 32-41, 43-47, 49, and 53 stand rejected under 35 U.S.C. § 101 as directed to non-statutory subject matter.

Regarding claims 32, 33, and 35-38 the Office Action rejects these claims stating, “The recited components of the claims can reasonably be interpreted as computer program modules—software per se.” Accordingly, the preambles of these claims have been amended to recite “A computer-readable storage medium containing a data structure ..., the data structure comprising:” Applicants note that data structure claims, often referred to as “Lowry Claims,” constitute valid statutory subject matter

pursuant to *In re Lowry*, 32 F.3D 1579 (Fed. Cir. 1994). In fact, the Lowry patent issued on September 2, 1997 as US Patent No. 6,353,829 and includes data structure claims (e.g., see independent claim 20 of 6,353,829).

Claims 41, 45-47, 50, and 53 have been amended to recite “computer readable **storage medium**” and paragraph [0014] of the originally filed specification has been amended to remove reference to propagated signals.

Claim Rejections – 35 U.S.C. § 102

Claims 1, 3-10, 16-20, 32-46, 50, and 53 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Goebel (US 6,289,505).

A claim is anticipated only if each and every element of the claim is found in a single reference. M.P.E.P. § 2131 (citing *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628 (Fed. Cir. 1987)). “The identical invention must be shown in as complete detail as is contained in the claim.” M.P.E.P. § 2131 (citing *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226 (Fed. Cir. 1989)).

Independent Claims 1, 18, 32, 41, and 50

Amended independent claim 1 now recites, in pertinent part,

executing the intermediate code based on external execution input;
generating data that indicates performance of the intermediate code **when**
the intermediate code is executed with the external execution input; and

Applicants respectfully submit that *Goebel* fails to disclose executing intermediate code based on external execution input. Applicants note that independent claim 1 has been amended to include the subject matter of dependent claim 16. Furthermore, similar subject matter is recited in original non-amended independent claim 32. As such, no new search is necessary.

Goebel disclose a technique to optimize an existing binary executable 313. The binary executable 313 (see FIG. 3 of *Goebel*) is disassembled by binary disassembler 311 to generate an intermediate representation (IR). The IR is then normalized by IR normalization segment 315. The normalized IR is input into the IR optimizer segment 305 where it is optimized and then processed by the code generator segment 307 to make a binary module 309 that can be linked to create a binary executable (e.g., binary

executable 313). IR optimizer 305 can also use profile information 317 to determine what portions of a binary executable need to be optimized. *Goebel*, col. 6, lines 45-61; col. 7, lines 3-51.

However, *Goebel* fails to disclose executing the IR based on external execution input. Rather, *Goebel* discloses that the IR is optimized by optimizer 305 based on profile information 317, **which is generated during execution of a binary executable**. In fact, several portions of *Goebel* explicitly state that profile information 317 is generated based on execution of a binary executable—not execution of the IR with external execution input. Some of these portions of *Goebel* are presented below.

Execution profile--An execution profile is a collection of data, gathered while a **program executes**, that reveals (without limitation) which procedures in the program are most frequently executed, the execution flow of the procedures, memory cache utilization, and other information that can be used to analyze the program's performance. **The execution profile is often obtained by inserting instrumentation procedures within a binary executable**. (*Goebel*, col. 4, line 66 to col. 5, line 6)

The intermediate representation optimizer segment 305 of the re-optimizing compiler 300 can also process **profile information 317 generated during execution of an instrumented binary executable** to determine which portions of the binary executable most need to be optimized. (*Goebel*, col. 6, lines 61-66)

These profiling instructions are used to gather and save profile data relating to the execution history and/or the memory cache performance of the **executing program**. **That is, when the binary executable is executed by the computer**, the profiling procedures measure relevant characteristics of the **executing program** and store this information as profile data. (*Goebel*, col. 7, line 63 to col. 8, line 2)

It is noteworthy, that *Goebel* clearly distinguishes between binary executable 313, which it explicitly defines as,

Binary executable--A binary executable is the data that is loaded into a computer's memory and that is executed by the computer's CPU,

and an intermediate representation (IR), which it explicitly defines as,

Intermediate representation (IR)--The intermediate representation is the representation of a source program that results after the source program has been processed by a compiler's front-end segment. The intermediate representation represents the structures and operations described in the source

program but in a form that is efficiently processed in subsequent segments of the compiler,

Accordingly, *Goebel* discloses executing a program or binary executable (e.g., binary executable 313) to generate profile information 317 to be used for optimizing an intermediate representation (IR). However, *Goebel* fails to disclose, teach, or suggest executing the IR based on external execution input to generate data, which is subsequently used to create machine code. **In short, *Goebel* does not disclose that IR optimizer 305 executes the IR based on external execution input, but rather receives profile information 317 that was generated by executing an instrumented version of binary executable 313 on a particular data set. Executing binary executable 313 on a particular data set does not disclose, teach, or suggest executing the IR on a particular data set.**

Consequently, *Goebel* fails to disclose each and every element of amended claim 1, as required under M.P.E.P. § 2131. Independent claims 18, 32, 41, and 50 include similar novel elements as independent claim 1. Accordingly, withdrawal of the instant §102 rejections of claims 1, 18, 32, 41, and 50 is requested.

Independent Claim 35

Independent claim 35 recites, in pertinent part,

a back-end code generator, coupled with the front-end code generator, to receive the intermediate code, produce machine code based on the intermediate code, and determine whether to produce the modified machine code, and **if the modified machine code is to be produced:**

the back-end code generator to provide the modified machine code to the profiler for generation of a second data file;

the front-end code generator to receive the second data file and produce second intermediate code based directly on the source code and based on the second data file; and

the backend generator to produce further modified machine code based on the second intermediate code.

Applicants respectfully submit that *Goebel* fails to disclose a front-end code generator coupled that produces second intermediate code based directly on the source code and a second data file.

To be sure, with reference to FIG. 3 of *Goebel*, the Examiner cites compiler front end 303 as corresponding to the claimed “front-end code generator,” cites source

information 301 as corresponding to the claimed “source code,” and cites profile information 317 as corresponding to the claimed “data file.” However, *Goebel* discloses a “capability to re-optimize a binary executable, originally not optimized, partially optimized, or optimized for a particular computer system, so that the binary executable is optimized for another targeted computer.” *Goebel*, col. 3, lines 30-35.

Goebel discloses that the **first instance** of intermediate code and binary module 309 are generated based on source information 301; however, **subsequent re-optimizations of binary executables** do not re-optimize their corresponding intermediate representations based directly on source information 301. Rather, *Goebel* discloses that binary executable 313 is disassembled into an intermediate representation by binary disassembler 311 and re-optimized by intermediate representation optimizer 305 using profile information 317. **The re-optimization is not based directly on source information 301.**

Consequently, *Goebel* fails to disclose each and every element of amended claim 35, as required under M.P.E.P. § 2131. Accordingly, withdrawal of the instant §102 rejections of claim 35 is requested.

The dependent claims are novel over the prior art of record for at least the same reasons as discussed above in connection with their respective independent claims, in addition to adding further limitations of their own. Accordingly, Applicants respectfully request that the instant § 102 rejections of the dependent claims be withdrawn.

Claim Rejections – 35 U.S.C. § 103

Claims 21, 22, 25, 26, 28-31, 46, 47, and 49 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Goebel* in view of *Shupak* (US 6,874,140).

Claims 23 and 24 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Goebel* in view of *Shupak* in further view of Applicant’s Admitted Prior Art (*AAPA*).

“To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. All words in a claim must be considered in judging the patentability of that claim against the prior art.” M.P.E.P. § 2143.03.

Amended independent claims 21 and 46 both now recite, in pertinent part,
produce the further modified intermediate and machine code **based on
the source code** and the another data file

For the reasons discussed above in connection with independent claim 35, Applicants respectfully submit that *Goebel* fails to disclose, teach, or suggest producing further modified intermediate code based on source code. Rather, *Goebel* discloses that the source code is only used for creation of the first instance of intermediate code—not subsequent modified instances. *Shupak* also fails to teach or suggest this element. Accordingly, the combination of *Goebel* and *Shupak* fails to teach or suggest this element, as well

Consequently, the combination of *Goebel* and *Shupak* fails to teach or suggest all elements of claims 21 and 46, as required under M.P.E.P. § 2143.03. Accordingly, Applicants request that the instant §103(a) rejections of claims 21 and 46 be withdrawn.

The dependent claims are nonobvious over the prior art of record for at least the same reasons as discussed above in connection with their respective independent claims, in addition to adding further limitations of their own. Accordingly, Applicants respectfully request that the instant § 103 rejections of the dependent claims be withdrawn.

CONCLUSION

In view of the foregoing amendments and remarks, it is believed that the applicable rejections have been overcome and all claims remaining in the application are presently in condition for allowance. Accordingly, favorable consideration and a Notice of Allowance are earnestly solicited. The Examiner is invited to telephone the undersigned representative at (206) 292-8600 if the Examiner believes that an interview might be useful for any reason.

CHARGE DEPOSIT ACCOUNT

It is not believed that extensions of time are required beyond those that may otherwise be provided for in documents accompanying this paper. However, if additional extensions of time are necessary to prevent abandonment of this application, then such extensions of time are hereby petitioned under 37 C.F.R. § 1.136(a). Any fees required therefore are hereby authorized to be charged to Deposit Account No. 02-2666. Please credit any overpayment to the same deposit account.

Respectfully submitted,

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

Date: July 2, 2007

/Cory G. Claassen/

Cory G. Claassen
Reg. No. 50,296
Phone: (206) 292-8600

1279 Oakmead Parkway
Sunnyvale, California 94085-4040

CERTIFICATE OF TRANSMISSION

I hereby certify that this correspondence is being submitted electronically via EFS Web on the date shown below to the United States Patent and Trademark Office.


Suzanne Johnston

7/2/07
Date